1A2 – Information structurée [1]

Lionel Vaux

IREM & Institut de Mathématiques de Marseille, Université d'Aix-Marseille

Formation ISN, niveau 1
Gardanne, 17 décembre 2014

http://isn.irem.univ-mrs.fr/wiki

1. Transparents quasi-intégralement issus de ceux d'Emmanuel Beffara en 2013

Plan

Structure de l'information
- Exemple 1 : une image
- Exemple 2 : une page de tableur
- Exemple 3 : les systèmes de fichiers

Données en programmation
- Structures de données
- Types élémentaires
- Types composés

La notion de type
- Typage en programmation
- Formats de fichiers

Dans l'épisode précédent

- on peut représenter des informations par des suites de bits
- il est nécessaire de connaître le codage employé par une suite de bits pour l'interpréter et l'utiliser.

Types et formats

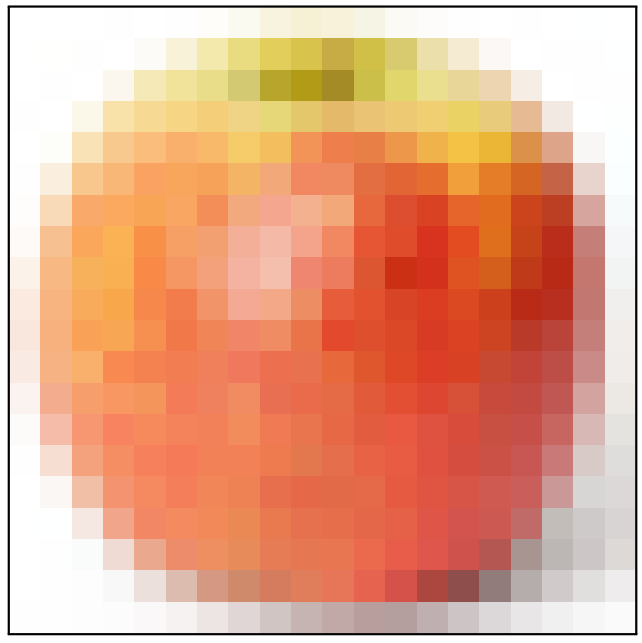
Les données numériques sont agencées de manière à en permettre le stockage et le traitement. L'organisation des données numériques respecte des formats qui sont soit *ad-hoc*, soit des conventions, soit des normes.

Cette nécessité de structuration est présente partout :

- pour les données dans la mémoire de l'ordinateur ;
- pour les variables employées dans les calculs ;
- pour les données enregistrées dans des fichiers ;
- pour le stockage, l'archivage, la communicationm la recherche de données…

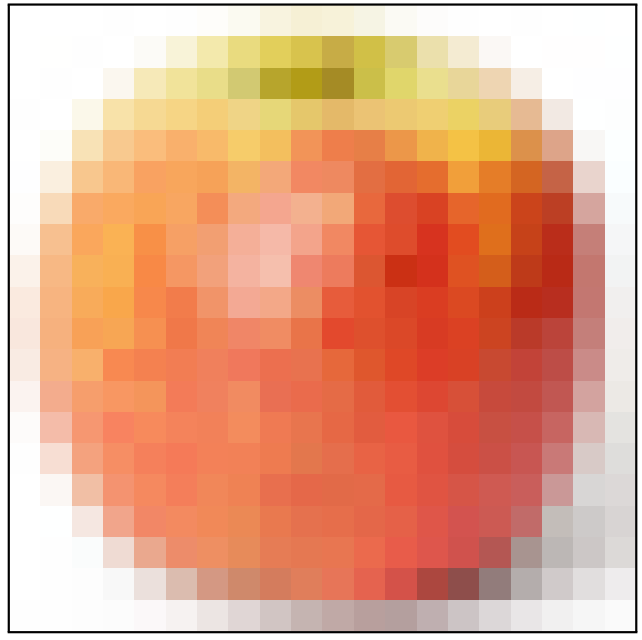
Exemple 1 : une image en couleurs

Exemple 1 : une image en couleurs



composée de pixels…

Exemple 1 : une image en couleurs



- Dimensions : 20×20 pixels.
- Pixels composés de rouge, vert, bleu.
- Pixel en (0,0) : blanc = rouge, vert, bleu à 100%
- Pixel en (13,7) : rouge à 84%, vert à 20%, bleu à 12%
- Informations supplémentaire (infos de prise de vue, etc.)

Décomposition

- Format :
  - Largeur : nombre entier (ici 20)
  - Hauteur : nombre entier (ici 20)
  - Nombre de plans : nombre entier (ici 3 : rouge, vert, bleu)
  - Valeur maximum des composants : entier (par exemple 255)
- Données “brutes” : liste des composantes des points
  - point (0,0) :
    - rouge : entier (255)
    - vert : entier (255)
    - bleu : entier (255)
  - point (1,0) : …
  - …
- Métadonnées :
  - “titre” → “pomme Pink Lady”
  - “temps de pause” → 1/20 s
  - “flash activé” → non
  - “date de prise” → 8/11/2010 à 15h27
  - etc.

Le format PPM (portable pixmap)

identifie le format

dimensions

valeur maximum des composantes

partie verte du 6ème pixel de la 1ère ligne

pink_lady.ppm

```
P3
20 20
255
255 255 255 255 255 255 255 255 255 254 254 254 255 255 255 254 254 254 254 253 250 251 250
241
248 243 223 246 240 213 247 242 217 246 244 228 251 250 244 253 252 251 254 254 254 255 255
255
254 254 254 255 255 255 254 255 255 255 255 255
255 255 255 255 255 254 254 254 254 255 255 255 252 251 247 248 243 216 243 233 172 233 219
127
226 207 91 216 195 77 199 172 70 209 192 71 216 203 111 235 224 172 244 235 210 251 248 245
255 255 255 255 254 254 255 254 254 254 255 255
255 255 255 254 254 254 255 255 255 251 247 239 246 233 184 241 227 154 233 221 137 210 201
114
183 165 44 178 156 23 165 139 37 203 191 73 224 214 107 234 223 142 232 214 153 238 213 178
246 238 229 255 255 255 254 254 255 254 254 255
254 254 254 255 255 255 251 248 234 248 226 169 246 217 147 246 213 133 244 206 120 239 212
134
230 215 120 229 199 107 228 185 105 235 195 116 236 201 114 239 207 114 235 210 100 232 204
123
230 186 147 242 233 227 254 255 255 253 254 254
255 255 255 253 253 249 248 226 182 248 201 143 250 189 124 249 176 108 248 186 106 246 204
106
243 191 94 242 148 88 238 126 75 231 127 71 236 151 73 240 179 75 244 194 69 235 182 54
220 145 75 221 164 137 248 247 245 252 255 255
...
```

Ce format ne contient pas de métadonnées.

D'autres formats

La même image pourrait être représentée par d'autres formats de fichiers correspondant à d'autres façons de décomposer et de représenter l'information :

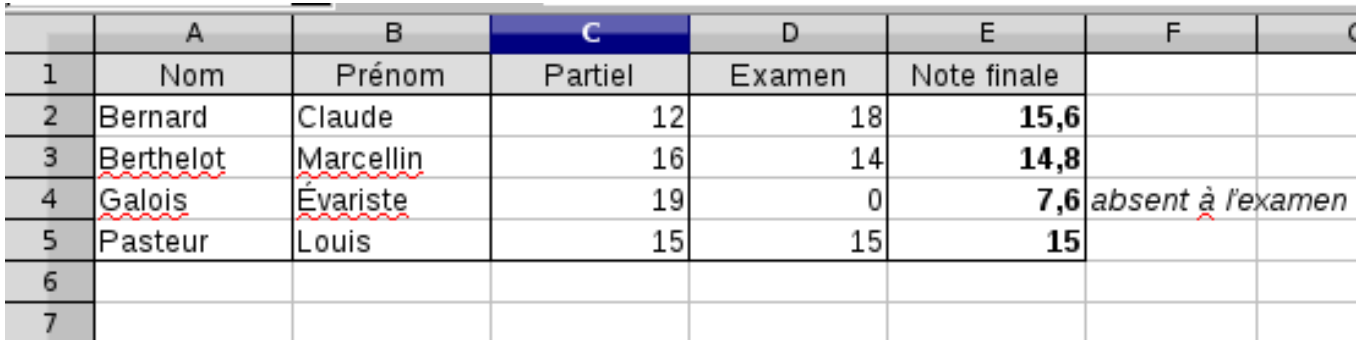
PNG format similaire, avec une étape de compression, peut contenir des métadonnées

BMP, GIF, PCX d'autres formats (propriétaires) pour dire essentiellement la même chose

JPEG format plus complexe, compression de l'image avec perte,
peut contenir des métadonnées (cf. EXIF utilisé par les appareils photo)

Exemple 2 : une page de tableur

Voici une feuille de notes :



	A	B	C	D	E	F
1	Nom	Prénom	Partiel	Examen	Note finale	
2	Bernard	Claude	12	18	**15,6**	
3	Berthelot	Marcellin	16	14	**14,8**	
4	Galois	Évariste	19	0	**7,6**	*absent à l'examen*
5	Pasteur	Louis	15	15	**15**	
6						
7						

Vous reconnaissez ce que l'on voit quand on utilise un tableur. On peut se demander comment représenter cette information dans un ordinateur.

Décomposition

Plusieurs façons possibles pour décrire cette feuille :

- Une ligne de titre suivie d'une suite de lignes de même nature.
- Une série de colonnes avec des données d'un seul type (mise à part la ligne de titre) :
  - La première colonne : une chaine de caractères
  - La seconde colonne : une chaine de caractères
  - La troisième colonne : un nombre entier
  - La quatrième colonne : un nombre entier
  - La cinquième colonne : un nombre à virgule flottante
  - La sixième colonne : une chaine de caractères
- La cinquième colonne calcule une moyenne : elle contient en fait une formule.
- Enfin il y a une certaine présentation : certaines cases sont en gras, certaines sont centrées…

Le format CSV (comma-separated values)

C'est un format textuel pour coder des données tabulaires.
Un fichier à ce format pourrait être :

notes.csv

```
Nom;Prénom;Partiel;Examen;Note finale;
Bernard;Claude;12;18;15,6;
Berthelot;Marcellin;16;14;14.8;
Galois;Évariste;19;0;7.6;absent à l'examen
Pasteur;Louis;15;15;15;
```

- Pratique : simple à produire et analyser, manipulable à la main
- Très pauvre :
  - seulement des chaînes de caractères et des nombres (avec des problèmes)
  - aucune présentation
  - aucune formule

Le format ODS (OpenDocument spreadsheet)

notes.ods

```
<office:spreadsheet>
  <table:table table:name="Sheet1" table:style-name="ta1">
   <table:table-column table:style-name="co1" table:number-columns-repeated="5" table:default-cell-style-na
    <table:table-row table:style-name="ro1">
    <table:table-cell table:style-name="ce1" office:value-type="string" calcext:value-type="string">
        <text:p>Nom</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce1" office:value-type="string" calcext:value-type="string">
        <text:p>Prénom</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce1" office:value-type="string" calcext:value-type="string">
        <text:p>Partiel</text:p>
      </table:table-cell>...
    </table:table-row>
    <table:table-row table:style-name="ro1">
    <table:table-cell table:style-name="ce2" office:value-type="string" calcext:value-type="string">
        <text:p>Bernard</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce2" office:value-type="string" calcext:value-type="string">
        <text:p>Claude</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce2" office:value-type="float" office:value="12" calcext:value-type
        <text:p>12</text:p>
      </table:table-cell>...
    </table:table-row>...
  </table:table>
</office:spreadsheet>
```

Le format ODS (OpenDocument spreadsheet)

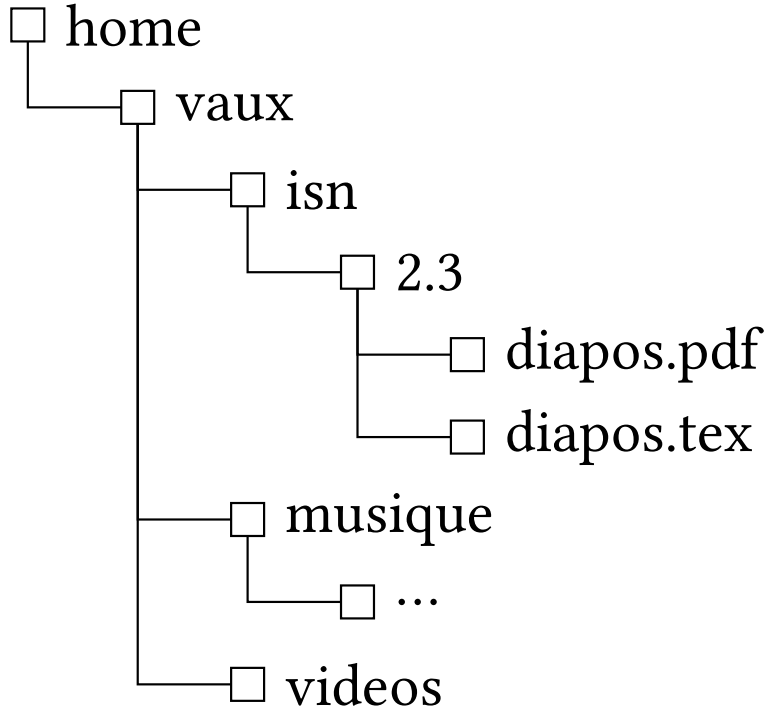
notes.ods

```
<office:spreadsheet>
  <table:table table:name="Sheet1" table:style-name="ta1">
   <table:table-column table:style-name="co1" table:number-columns-repeated="5" table:default-cell-style-na
    <table:table-row table:style-name="ro1">
    <table:table-cell table:style-name="ce1" office:value-type="string" calcext:value-type="string">
        <text:p>Nom</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce1" office:value-type="string" calcext:value-type="string">
        <text:p>Prénom</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce1" office:value-type="string" calcext:value-type="string">
        <text:p>Partiel</text:p>
      </table:table-cell>...
    </table:table-row>
    <table:table-row table:style-name="ro1">
    <table:table-cell table:style-name="ce2" office:value-type="string" calcext:value-type="string">
        <text:p>Bernard</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce2" office:value-type="string" calcext:value-type="string">
        <text:p>Claude</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce2" office:value-type="float" office:value="12" calcext:value-type
        <text:p>12</text:p>
      </table:table-cell>...
    </table:table-row>...
  </table:table>
</office:spreadsheet>
```

En réalité c'est un peu plus compliqué…

Exemple 3 : les systèmes de fichiers

Pour organiser ses données informatisées, on les stocke dans un système de fichiers, structuré en une arborescence de répertoires.



Chaque fichier contient une suite de bits, destinée à être interprétée d'une certaine façon…

Pour résumer

Une unité d'information donnée est en général composée de nombreuses parties, il y a donc de nombreuses façons de la représenter, en fonction de plusieurs facteurs :

modélisation Quelles sont les informations du monde réel que l'on souhaite prendre en compte ?

structuration Une fois les informations identifiées, comment les décompose-t-on ?
Quelles sont les briques de base ?

codage Quelle est la représentation utilisée pour représenter concrètement l'information structurée ?

Il y a en général de nombreux choix pertinents possibles.

Structuration des données

En analysant l'information à manipuler, on identifie

- des façons de décomposer les données complexes :
  - structures fixes (couple largeur / hauteur)
  - collections linéaires (la suite des pixels)
  - collections de données nommées (les métadonnées)
  - structures plus complexes (arbres, graphes…)
- des types de données qui ne se décomposent plus :
  - nombres (entiers ou pas)
  - valeurs de vérité
  - caractères
  - …

Structures de données

Tentative de définition

Une structure de données est une façon d'organiser des données afin de rendre possible (et efficace) leur mémorisation et leur traitement.

Structures de données – exemples

- Les constantes ou les variables (entières, flottantes, booléennes, caractères, pointeurs, etc).
- Les chaînes de caractères.
- Les tableaux à une ou plusieurs dimensions (vecteurs, matrices, etc) indicés par des entiers ; de taille fixe ou variable.
- Les tableaux associatifs, indicés par des clés (des chaînes de caractères).
- Les enregistrements (groupes de structures de données).
- Les structures récursives (listes, arbres, graphes, ...).
- Les bases de données.

Il existe de nombreuses structures bien connues (parfois très complexes) ; on peut en inventer à l'infini, selon les problèmes à traiter.

Structures de données – leur importance

On choisit une structure de données pour modéliser un problème, en fonction de différents critères, parfois contradictoires :

- simplicité de conception des algorithmes ;
- minimisation du coût en mémoire ;
- efficacité en accès, insertion ou modification ;
- vitesse de calcul ;
- efficacité pour l'architecture d'une machine ;
- contrôle des données ;
- disponibilités d'outils (langages, bibliothèques).

Types élémentaires (ou simples)

Ils sont donnés comme éléments de base, on n'accède pas à leur structure.

Ils peuvent être fournis par

le processeur : entiers de taille fixe, nombres à virgule flottante

le langage de programmation : booléens, caractères, chaînes de caractères…

des bibliothèques : nombres de précision arbitraire, vecteurs, images…

Ils dépendent donc du langage de programmation et des outils utilisés !

Types élémentaires – exemples

Ils dépendent du langage, leur représentation et leur taille peuvent dépendre de la machine :

Type simple	C		Java		Python	
entier court	`short`	16	`short`	16		
entier	`int`	m	`int`	32	–	
entier long	`long`	m	`long`	64		
entier double	`long long`	$2m$				
grand entier	(bib. GMP)		`BigInteger`		`int`	
flottant simple	`float`	32	`float`	32		
flottant double	`double`	64	`double`	64	`float`	64
booléen	`int`	m	`boolean`	8	`bool`	
caractère	`char`	8	`char`	16	`str`	
chaîne	`char*`		`String`		`str`	

(les tailles sont en bits ; m = taille d'un mot machine)

Opérateurs

Les types simples et certains types composés sont associés à des opérateurs ; ils dépendent du langage. Certaines opérations sont des opérateurs dans un langage et des fonctions dans d'autres.

Opérateurs	**Types**	**C**	**Java**	**Python**
+ − ×	ent., flottants	`+ - *`	`+ - *`	`+ - *`
div réelle	flottants	`/`	`/`	`/`
div entière	entiers	`/ %`	`/ %`	`// %`
puissance	ent., flottants	`pow()`	`Math.pow()`	`**`
concat.	chaînes	`strcat()`	`+`	`+`
binaires	entiers	`~ & |`	`~ & |`	`~ & |`
logiques	booléens	`! && ||`	`! && ||`	`not and or`
égalités	simples	`== !=`	`== !=`	`== !=`
inégalités	simples	`< <=`	`< <=`	`< <=`

Types composés

Il s'agit uniquement de constructions propres aux langages de programmation : les processeurs ne les manipulent pas directement. Ils peuvent être

- fournis par le langage
  (tableaux en C, listes et dictionnaires en Python, chaînes de caractères)
- fournis par des bibliothèques
  (structures complexes ou optimisées, bases de données)
- construits au moyen des possibilités du langage
  (tableaux, n-uplets, enregistrements, classes…)

Tableaux à une dimension

- C'est une suite de cases du même type (n'importe lequel),
- accessibles par un indice entier.
- Le tableau peut avoir une capacité, fixe ou variable, et une longueur courante.

Exemple : un tableau d'entiers

1	1	2	3	5	8	13	21	34	?	?	?
0	1	2	3	4	5	6	7	8	9	10	11

Dans divers langages de programmation :

C	Java	Python
`[]`	`[]`, `Array`	`array.array`

Tableaux à une dimension – exemples

Tableau de chaînes :

"Bonjour"	"les"	"amis"	?	?	?
0	1	2	3	4	5

Tableau de tableaux :

0	10	0	20	0	77	0	60
1	15	1	22	1	50	1	20
2	20	2	50	2	0	2	60
3	25	3	52	3	-5	3	80
0		1		2		3	

C'est *une* façon de représenter les tableaux en plusieurs dimensions (matrices, images, ...).

Chaînes de caractères

Permet de mémoriser du texte : `"Bonjour les amis"`

- C'est une suite de caractères,
- accessibles par un indice entier.

Formellement, cela ressemble donc à un tableau de caractères.
Conceptuellement, c'est différent.

Les opérations courantes sont :

- calcul de longueur,
- comparaison (alphabétique),
- concaténation,
- extraction ou recherche d'une sous-chaîne…

Dans divers langages de programmation :

C	Java	Python
`char*`	`String`	`str`

Chaînes de caractères – représentation

Il y a principalement deux modes de représentation :

À la C un tableau de caractères de longueur indéterminée, avec un marqueur de fin (en général le caractère de code 0)

À la Pascal un tableau de caractères et une longueur explicite

Les caractères utilisent un codage particulier :

- C : un code sur 8 bits non spécifié par le type
  parfois ASCII, parfois ISO-8859-1, parfois UTF-8…
- Java : UCS-2 = code sur 16 bits selon Unicode
- Python 2 : soit à la C (`str`) soit à la Java (`unicode`)
- Python 3 : à la Java (`str`)
  il y a aussi un type de suites d'octets : `bytes`

Chaînes immuables

Dans certains langages, les chaînes sont immuables (Java, Python) :

```
>>> s = "bonjour"
>>> s
'bonjour'
>>> type(s)
<type 'str'>
>>> s[1]
'o'
>>> s[1] = 'x'
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
>>>
```

Dictionnaires

Un dictionnaire (ou tableau associatif, ou *map*) associe un ensemble de clés à un ensemble de valeurs.

- C'est une généralisation des tableaux, les indices entiers étant remplacés par des chaînes de caractères (ou tout autre type de donnée adapté).
- Les opérations standard sont :
  - ajout ;
  - modification ;
  - suppression ;
  - recherche.
- Pour être efficaces, ils sont implémentés en utilisant des tables de hachage ou des arbres équilibrés.

Dictionnaires en Python

```
>>> d = {'Nom' :'John', 'Prénom' :'Doe', 'Age' :17}
>>> d
{'Nom': 'John', 'Age': 17, 'Prénom': 'Doe'}
>>> d['Age']
17
>>> d['Notes'] = [18.5, 14, 16.5]
>>> d
{'Nom': 'John', 'Age': 17, 'Prénom': 'Doe', 'Notes':
[18.5, 14, 16.5]}
>>> d['Notes'][1]
14
>>> del d['Notes']
>>> d
{'Nom': 'John', 'Age': 17, 'Prénom': 'Doe'}
```

Enregistrements

- C'est un groupe de structures de données (les champs),
- pouvant être de types différents,
- en nombre fixé,
- identifiées par un nom de champ ;
- les noms de champs sont immuables.

Exemple :

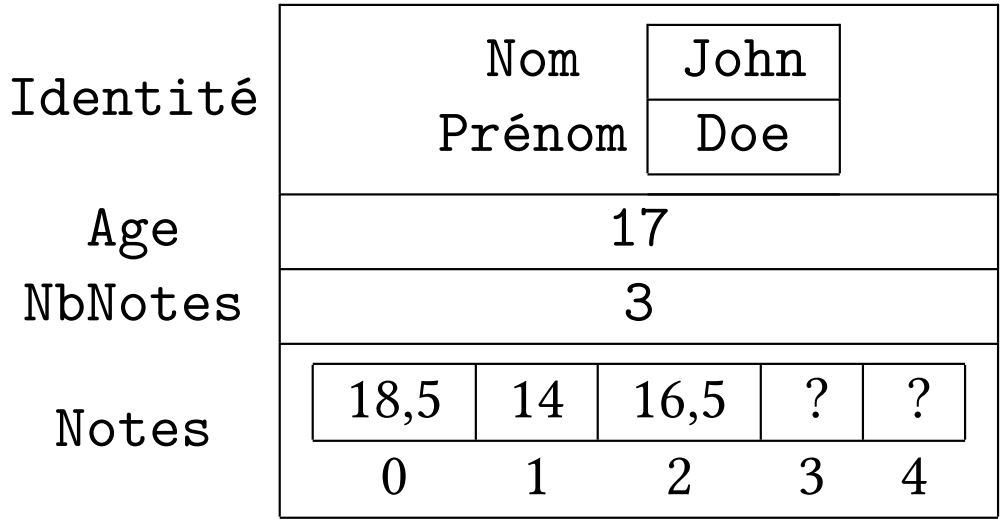
`Nom`	`John`
`Prénom`	`Doe`
`Age`	`17`

Dans divers langages de programmation :

C	Java	Python
`struct`	`class`	`dict` ou classes

Enregistrements – exemple

Les champs d'un enregistrement peuvent être des tableaux, des enregistrements, etc.



Un mot sur les structures récursives

Il s'agit de structures de données de taille variable, définies de façon récursive.

- Listes chaînées :
  - Une liste est soit la liste vide, soit un élément de tête suivi d'une liste (la queue) ;
  - c'est efficace pour insérer des éléments,
  - l'accès à un élément par son numéro nécessite de parcourir les précédents.
- Arbres :
  - un ensemble de cellules ayant chacune un parent (sauf la racine) et un certain nombre de descendants
  - utilisés pour manipuler efficacement de grandes quantités de données variables : arbres de recherche, tas, B-trees…

La notion de type

Le mot *type* désigne donc plusieurs choses :

- un ensemble de données que l'on peut représenter,
- un ensemble d'opérations que l'on peut effectuer sur ces données,
- une façon de décomposer et structurer ces données,
- une façon de représenter concrètement ces structures.

Selon le contexte, tout ou partie de ces éléments sont concernés.

Types concrets

Il s'agit des types pour lesquels on donne explicitement la structure des données.
Exemples :

- entiers sur 64 bits
- tableaux
- listes chaînées
- dictionnaires représentés par des arbres de recherche

On parle aussi de *types de données*.

Types abstraits

Il s'agit de types pour lesquels on ne donne pas la structure mais uniquement les opérations disponibles, avec des garanties (effet, coût) sur le comportement de ces opérations.
Exemples :

- entiers de précision arbitraire
- collections séquentielles avec insertion
- associations clés/valeurs
- bases de données

Ils permettent de gagner en *modularité*.

Formats de fichiers

Il s'agit du niveau le plus concret dans la spécification d'un type : la représentation d'une donnée au moyen d'une suite de bits, pour la stocker sur un support physique.

- Le format d'un fichier est l'organisation physique des données à l'intérieur de celui-ci ; c'est une structure de données, et une façon de la coder.
- Le format d'un fichier
  - découle d'une norme, ou d'un standard de fait ;
  - il peut être convert par un secret industriel, mais cela ne favorise pas l'interopérabilité (cf. rétro-ingéniérie)

Degrés de standardisation

Les structures de données sont souvent construites avec les mêmes ingrédients :

- nombres, chaînes de caractères, dates, etc.
- collections ordonnées
- enregistrements et dictionnaires

Pour cette raison il existe des formats génériques pour représenter toute donnée exprimée dans ces structures :

- données structurées représentées textuellement : CSV, XML, JSON…
- conteneurs pour arborescences de fichiers, avec compression : Zip et ses dérivés (ODT, JAR, …), Tar/GZip…

Cela permet de réutiliser une partie du travail de codage et décodage d'un format à l'autre.

Le format ODS (OpenDocument spreadsheet)

notes.ods

```
<office:spreadsheet>
  <table:table table:name="Sheet1" table:style-name="ta1">
   <table:table-column table:style-name="co1" table:number-columns-repeated="5" table:default-cell-style-na
    <table:table-row table:style-name="ro1">
    <table:table-cell table:style-name="ce1" office:value-type="string" calcext:value-type="string">
        <text:p>Nom</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce1" office:value-type="string" calcext:value-type="string">
        <text:p>Prénom</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce1" office:value-type="string" calcext:value-type="string">
        <text:p>Partiel</text:p>
      </table:table-cell>...
    </table:table-row>
    <table:table-row table:style-name="ro1">
    <table:table-cell table:style-name="ce2" office:value-type="string" calcext:value-type="string">
        <text:p>Bernard</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce2" office:value-type="string" calcext:value-type="string">
        <text:p>Claude</text:p>
      </table:table-cell>
    <table:table-cell table:style-name="ce2" office:value-type="float" office:value="12" calcext:value-type
        <text:p>12</text:p>
      </table:table-cell>...
    </table:table-row>...
  </table:table>
</office:spreadsheet>
```

Le format ODS – encore simplifié

Il s'agit un format basé sur XML (Extensible Markup Language) que l'on pourrait simplifier ainsi :

```
<spreadsheet>
    <table>
        <row>
            <cell>..</cell>
            <cell>..</cell>
        </row>
        <row>
            <cell>..</cell>
            <cell>..</cell>
            <cell>..</cell>
        </row>
        <row>
             ...
        </row>
    </table>
</spreadsheet>
```

Le format ODS – un mot sur XML

Ce format XML (plus général que HTML) sert a représenter de l'information structurée. Il y a la notion d'attribut pour la mise en forme.

On peut dire que :

- un classeur est une suite de tables ;
- une table est une suite de rangées ;
- une rangée est une suite de cellules.

Dans cette description on ne voit pas directement la notion de table avec un nombre de colonnes et de lignes fixées : ce n'est pas la représentation d'un tableau rectangulaire.

On sépare de manière claire l'information d'un côté et sa présentation de l'autre côté. Cette présentation est à son tour un objet que l'on peut également représenter en XML.

Le format ODS – interopérabilité

Si tout le monde se met d'accord sur le sens du précédent format, alors on peut travailler avec le même document sur différents systèmes, sur différentes machines et dans différentes langues. c'est l'interopérabilité.

ODS fait partie d'une norme : OASIS Open Document Format for Office Applications (odf)

XLSX est une extension de nom de fichier pour tableur au format Office Open XML = norme ISO/IEC (IS 29500) créée par Microsoft. Même type de format XML que ODS.

Le fait d'avoir 2 vraies normes permet relativement facilement de passer de l'une à l'autre : Excel permet de lire des documents ODS et LibreOffice (OpenOffice) permet de lire des documents XLSX.

Même histoire pour ODT

Un Titre

Voici un **texte** *simple*.

Et un autre paragraphe.

```
 <office:body>
  <office:text>
  <text:h text:style-name="Heading_20_2" text:outline-level="2">
       Un Titre
   </text:h>
   <text:p text:style-name="P2">
      Voici un
      <text:span text:style-name="T1">
          texte
      </text:span>
      <text:span text:style-name="T2">
          simple
      </text:span>.
   </text:p>
   <text:p text:style-name="P2"/>
   <text:p text:style-name="P2">
       Et un autre paragraphe.
    </text:p>
   <text:p text:style-name="P1"/>
  </office:text>
 </office:body>
</office:document>
```

Formats de fichiers classiques

Catégorie	**Formats**
Images	PNG, TIFF, JPEG, GIF, BMP
Dessin vectoriel	SVG, EPS
Son	OGG, FLAC, MP3, WAV, WMA, AAC
Vidéo	MPEG, OGM(DVD, DivX, XviD), AVI, Theora, FLV, MP4, MKV
Page	PDF, PostScript, HTML, XHTML
Document de bureautique	ODT, TXT, DOC(X), RTF, ODS, XLS(X), ...
Archives (fichier compressé)	7Z, TAR, GZIP, ZIP, LZW, RAR